

REMARKS

Entry of the above amendment is respectfully requested.

The new claims 1-11 are substantially similar to original claims 1-11 set out in the PCT International Application, with the exception that the claims have been amended to conform to U.S. practice. The Examiner is kindly requested to renumber the new claims 1-11 as claims 12-22 respectively, and to change the dependency accordingly.

Respectfully submitted,
PIPER MARBURY RUDNICK & WOLFE

Date: July 6, 2000

By:


Michael L. Kenaga
Reg. No. 34,639

PIPER MARBURY RUDNICK & WOLFE
P.O. Box 64807
Chicago, IL 60664-0807
(312) 368-4000

CLAIMS

1.. A method of compacting an intermediate programme consisting of a sequence of standard instructions, used in an on-board system, this on-board system being provided with a memory and a programme language interpreter capable of turning the intermediate programme into instructions of an object code that can be run directly by a microprocessor, said method consisting in:

a) searching through the intermediate programme for identical sequences of successive standard instructions;

b) subjecting the identical sequences of successive instructions to a comparison test to find a function, based on at least the number of occurrences of these sequences in said intermediate programme, that is higher than a reference value and, if the test returns a positive response, for each identical sequence of successive standard instructions which satisfies said test step,

c) a specific instruction is generated by defining a specific operating code and associating this specific operating code with the sequence of successive standard instructions which satisfied said test,

d) replacing each occurrence of each sequence of standard successive instructions in said intermediate programme with said specific operating code associated with it to obtain a compacted intermediate programme, consisting of a series of standard instructions and specific operating codes, and

e) storing in said memory an execution table which enables a reciprocal link to be made between each specific operating code inserted and the sequence of successive

standard instructions associated with the latter, which enables the memory space occupied by said compacted intermediate programme to be optimised by storing only one occurrence of said identical sequences of successive standard instructions in said memory.

5 2. A method as claimed in claim 1, characterised in that said function is also a function of the size of each identical sequence of successive instructions.

10 3. A method as claimed in claim 1, characterised in that in order to compress a plurality of intermediate programmes, said method also consists in:

15 - storing the execution table relating to at least one compacted intermediate programme and, for every additional intermediate programme subjected to a compaction process,

15 - reading said stored execution table and
15 - running the compaction process for every additional programme, taking account of the specific codes and instructions stored in this execution table.

20 4. A method of running a compacted intermediate programme obtained by applying the compaction method as claimed in claim 1, and consisting of a succession of standard instructions and specific operating codes stored in the memory of an on-board system, characterised in that it consists in:

25 - recognising in said memory the existence of a stored execution table containing at least one sequence of successive instructions associated with a specific operating code by means of a reciprocal link;

30 - calling up a command, via the interpreter, to read the successive standard instructions or specific operating codes of the compacted intermediate programme and, in the presence of a specific operating code:

30 - retrieving said sequence of successive

00200000000000000000000000000000

instructions associated with said specific operating code from the memory by means of a read instruction and, in the presence of a standard instruction,

- 5 - commanding the execution of this instruction by means of a read instruction,

5. A method as claimed in claim 4, characterised in that if a sequence of successive instructions associated with a specific operating code is called up, the current 10 value of a programme counter is incremented in a stack associated with the specific operating codes and a programme pointer points to the first instruction of said sequence of specific instructions, after which, on running an instruction to end the sequence of specific instructions, 15 said programme counter is decremented and the execution process continues starting with the next instruction or specific operating code.

6. A method as claimed in claim 5, characterised in that the stack associated with the specific operating codes 20 and the stack associated with the standard instructions are a single stack.

7. A multi-application on-board system comprising computing resources, a memory and language interpreter capable of turning an intermediate programme into 25 instructions which are directly executable by the computing resources, characterised in that said multi-application on-board system also has at least one table of standard codes constituting said intermediate programme stored at the level of said interpreter;

30 - at least one compacted intermediate programme constituting an application and consisting of a series of specific instruction codes and standard instruction codes, said specific instruction codes corresponding to sequences

of successive standard instructions;

- an execution table enabling a reciprocal link to be made between a specific operating code and the sequence of successive standard instructions associated with the latter,
 - 5 said at least one compacted intermediate programme and said execution table being stored in said memory, enabling the memory space occupied by said compacted intermediate programme to be optimised by storing in said programmable memory only one occurrence of said identical sequences of successive instructions.

8. An on-board system as claimed in claim 7, characterised in that said execution table comprises at least:

- a file of successive sequences corresponding to the specific instructions;

- a table of specific instruction codes and addresses at which these specific instructions are embedded in the table of successive sequences.

9. An on-board system as claimed in claim 8,
characterised in that said file of successive sequences
corresponding to the specific instructions and said table of
specific instruction codes are stored in a programmable
memory of said on-board system.

10. A compaction system for an intermediate programme,
this intermediate programme consisting of a series of
standard instructions which can be executed by a target
unit, characterised in that said system comprises at least:

- means for analysing all the standard executable instructions enabling, by means of a reading process, said intermediate programme to distinguish between and establish a list of all the sequences of executable standard instructions contained in this intermediate programme;
 - means for counting the number of occurrences in this

intermediate programme of each of the sequences of executable standard instructions forming part of this list;

- means for allocating to at least one sequence of executable standard instructions a specific code associated with this sequence of executable standard instructions in order to generate a specific instruction;

- means for replacing, in the programme, each occurrence of this sequence of executable standard instructions with the specific code associated with this sequence of executable standard instructions, representative of said specific instruction, which enables a compacted programme to be generated comprising a succession of executable standard instructions and specific instructions.

11. A system as claimed in claim 10, characterised in
that said means for allocating to at least one sequence of
executable standard instructions a specific code associated
with this sequence of executable standard instructions in
order to generate a specific instructions comprises at
least:

20 - means for computing the value of a function based on
at least the length of and number of occurrences of this
sequence of executable standard instructions, said function
being representative of the compression gain for this
sequence of executable standard instructions;

25 - means for comparing the value of this function with
a threshold value and, if said comparison returns a positive
response,

- means for writing to a file, with a reciprocal link,
a specific code and this sequence of executable standard
instructions in order to constitute said specific
instruction.